



Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>
Eprints ID : 17089

The contribution was presented at GTMG 2016 :
<https://gtmg2016.sciencesconf.org/>

To cite this version : Durix, Bastien and Morin, Géraldine and Chambon, Sylvie and Roudet, Céline and Garnier, Lionel and Cots, Olivier *Reconstruction basée squelette d'un objet 3D à partir de points de vue multiples*. (2016) In: Journées du Groupe de Travail en Modélisation Géométrique (GTMG 2016), 23 March 2016 - 24 March 2016 (Dijon, France).

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

Reconstruction basée squelette d'un objet 3D à partir de points de vue multiples

Bastien Durix¹, Géraldine Morin¹, Sylvie Chambon¹, Céline Roudet², Lionel Garnier², Olivier Cots¹

¹ IRIT, UMR CNRS 5505, Université Paul Sabatier, 31000 Toulouse

² LE2I, UMR CNRS 6306, Université de Bourgogne, 21000 Dijon

Résumé

L'avantage d'une reconstruction 3D basée squelette est de générer complètement un objet 3D à partir d'un ensemble de vues calibrées. Nous reconstruisons dans cet article des objets 3D dont le squelette est curviligne, à partir d'un nombre arbitraire de vues calibrées desquelles nous extrayons un squelette perspectif. Avoir plusieurs vues est nécessaire pour une reconstruction correcte, mais la projection d'un squelette peut modifier sa topologie. La contribution principale est d'estimer un squelette curviligne 3D, à partir de multiples images : sa topologie est choisie comme étant la plus proche des squelettes sur l'ensemble des images, ce qui signifie que le nombre de changements topologiques du squelette curviligne 3D aux images est minimisé. De cette façon, le nombre de branches appariées est maximal. Ensuite, nous introduisons une méthode de mise en correspondance pour la triangulation des branches 3D associées. Enfin, nous évaluons le conditionnement induit par les points de vue choisis, ce qui permet d'estimer une nouvelle vue à ajouter pour améliorer la reconstruction du squelette 3D de façon significative.

The advantage of skeleton-based 3D reconstruction is to completely generate a single 3D object from a set of calibrated views. Having numerous views is necessary for a reliable reconstruction but projections of skeletons lead to different topologies. We reconstruct 3D objects with curved medial axis from the perspective skeletons on an arbitrary number of calibrated acquisitions. The main contribution is to estimate the 3D skeleton, from multiple images : its topology is chosen as the closest to those of the perspective skeletons on the set of images, which means that the number of topology changes to map the 3D skeleton to images is minimum. This way, the number of matched branches is maximum. Then, we introduce a method to match each set of branches in order to triangulate the associated 3D branches. Finally, we evaluate the condition number corresponding to the viewpoints, which proposes a new viewpoint able to improve the reconstruction of the skeleton significantly.

Mots-clés : Modélisation géométrique, squelette, reconstruction, topologie

1. Introduction

Le squelette est un modèle de représentation des formes, introduit par Blum [Blu67]. À l'instar du squelette animal, il s'agit d'une structure, centrée à l'intérieur de la forme, dont chaque point correspond au centre d'une sphère maximale à l'intérieur de la forme (cf. Fig. 1). La structure a pour propriété de caractériser entièrement la forme.

Les squelettes sont notamment utilisés pour l'analyse de

formes [SSdBA11], la reconnaissance de formes [SSDZ99, XWLB], ou encore l'animation [LWFCML03, AHLD07, KCZO07]. Comme il s'agit d'une version simplifiée de la forme, l'analyse du squelette est plus aisée que l'analyse de la forme en elle-même. Dans cet article, nous allons l'utiliser pour reconstruire des objets à partir d'un ensemble d'images.

Dans le domaine de la reconstruction, sans parler de squelettes, de nombreuses approches ont été proposées : la reconstruction multi-vues [FP10] permet de retrouver un nuage de points représentant les parties visibles d'un objet sur un ensemble d'images, mais sans la topologie. Un autre type d'approche, le *Shape-from-Silhouette*, permet de retrou-

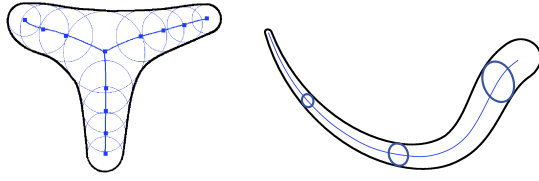


Figure 1: À gauche, représentation du squelette d'une forme, qui est l'ensemble des centres des boules maximales d'une forme. À droite, représentation d'un squelette perspectif, correspondant à un ensemble d'ellipses maximales, projections perspectives de sphères [DMC* 15a, DMC* 15b].

ver l'enveloppe visuelle d'un objet, qui est un volume implicite dans lequel l'objet est contenu [CBK03] : il s'agit de l'intersection des cônes généralisés contenant les différentes silhouettes projection de l'objet. Récemment, Chen *et al.* ont proposé une méthode de reconstruction à partir d'une seule image, basée sur les cylindres généralisés [CZS* 13]. Cette méthode identifie des objets 3D tubulaires le long d'un guide tracé par l'utilisateur, en considérant que la projection utilisée est orthographique.

Dans cet article, nous proposons une estimation du squelette curviligne 3D d'un objet réel, à partir d'un ensemble de projections. Cette approche a été employée par Livesu *et al.*, qui ont proposé une méthode d'estimation d'un squelette curviligne d'un modèle 3D connu [LGS12, LS13], à partir d'un ensemble de projections orthographiques. L'idée est d'estimer le squelette de chacune des projections de l'objet, puis de trianguler un ensemble de sphères (en utilisant la géométrie épipolaire pour associer les points des squelettes des projections). La topologie est ensuite ajoutée pour donner une cohérence au squelette curviligne 3D. Pour des raisons d'optimisation, les projections orthographiques sont fixées, ce qui ne permet pas d'appliquer la méthode pour des images perspectives.

Les travaux de Caglioti *et al.* proposent un algorithme d'estimation d'un squelette curviligne à rayon constant d'un objet, à partir d'une image perspective [CG06]. Leur approche considère une projection perspective de la forme sur une seule image. Comme il s'agit de la projection d'une surface canal à rayon constant, l'objet peut être reconstruit sans ambiguïté en appliquant une transformation homothétique à chaque sphère de la surface canal. Comme leur approche nécessite que l'objet ait un rayon constant, ceci limite la reconstruction à des objets spécifiques.

Enfin, dans un travail récent, nous avons proposé un algorithme permettant d'extraire le squelette perspectif associé à la projection d'une forme [DMC* 15a, DMC* 15b]. Ce squelette perspectif correspond à la projection d'un squelette 3D, en connaissant les paramètres de la caméra (*cf.* Fig. 1). Cet algorithme permet de retrouver la projection du squelette 3D dans une image perspective en s'affranchissant de la

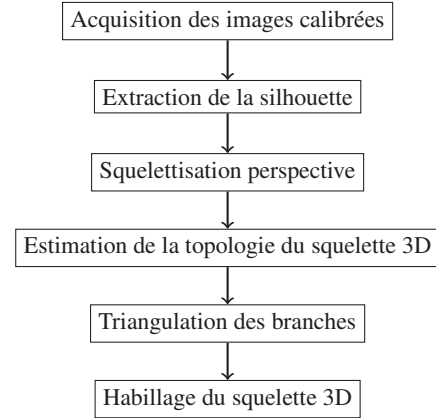


Figure 2: Représentation de l'enchaînement d'étapes de l'algorithme de reconstruction basée squelette.

contrainte de rayon constant, et d'estimer ainsi une reconstruction à partir de deux images. La finalité de la méthode est de reconstruire un squelette curviligne 3D, approximant au mieux l'objet 3D.

2. Description de l'algorithme

Nous présentons ici l'algorithme de reconstruction basée squelette, énoncé sur la Figure 2. Nous faisons le choix ici de n'utiliser que du matériel grand public, à savoir un appareil de capture de type *webcam* ou caméra de *smartphone*.

Cette méthode nécessite dans un premier temps d'acquérir des images calibrées d'un même objet, sous différents angles. Nous utilisons pour ceci un ensemble de marqueurs disposés autour de l'objet [KB99]. La seconde étape consiste à extraire la silhouette de l'objet sur chacune des images, avec l'utilisation de l'algorithme GrabCut [RKB04].

Le squelette perspectif de l'objet est ensuite estimé sur chacune des images. Les squelettes perspectifs correspondent à la projection du squelette curviligne 3D de l'objet sur les différentes images [DMC* 15a, DMC* 15b].

Vient ensuite l'étape d'estimation de la topologie du squelette 3D, conjointe à l'appariement des squelettes sur les différentes images : on obtient à l'issue de cette étape la branche de chaque squelette perspectif correspondant à une branche du squelette 3D.

Enfin, comme chaque branche est paramétrée, l'étape suivante est de mettre en correspondance les paramétrisations sur les branches des différents points de vue, afin de trianguler la branche 3D. La sortie de l'algorithme est un squelette 3D curviligne, qui peut être habillé pour obtenir un maillage 3D de l'objet.

Ainsi dans cet article, la première problématique étudiée est celle de l'appariement des branches de squelette sur les

différentes images. En effet, différents squelettes perspectifs d'un même objet peuvent avoir des topologies différentes, notamment à cause des auto-occultations de la forme (cf. Fig. 3). De fait, l'appariement des différentes branches des squelettes est rendu non trivial. Sachant l'on peut transformer un squelette en un autre par une succession de transformations topologiques, nous proposons dans la Section 3 une méthode permettant d'évaluer la topologie du squelette 3D, comme étant la plus proche, au sens des transformations topologiques, de l'ensemble des squelettes perspectifs.

Par la suite, nous étudions le procédé de triangulation d'une branche 3D de squelette. Une nécessité de la méthode est de mettre en correspondance les différentes paramétrisations des branches des squelettes projectifs. Nous proposons de modéliser cette mise en correspondance par l'estimation d'une géodésique, dans l'espace des paramétrisations, dans la Section 4.

Enfin, nous montrons que la mise en correspondance des paramétrisations des branches peut être ambiguë dans certains cas : en effet, il est tout à fait possible que plusieurs mises en correspondance différentes soient valables pour la triangulation d'une même branche 3D. Nous montrons dans la Section 5 comment détecter l'apparition de cette ambiguïté, et comment choisir une nouvelle vue pour la résoudre.

Ce travail est validé par des évaluations quantitatives, sur images de synthèse, et par des évaluations qualitatives sur images réelles.

3. Appariement topologique de squelettes

3.1. Mise en correspondance des squelettes

Bien que les squelettes perspectifs soient, en général, les projections du squelette 3D, ils n'ont pas nécessairement la même topologie (par exemple à cause des auto-occultations depuis les points de vue considérés). Prenons pour exemple l'objet de la Figure 3 : la topologie du squelette perspectif estimé sur la première image est différent de celles des squelettes estimés sur les deuxième et troisième images. De fait, leurs branches ne peuvent pas être appariées. Notre problématique ici sera de déterminer une topologie pouvant être appliquée au squelette 3D, et d'apparier les branches des squelettes perspectifs pour suivre cette topologie.

De nombreuses techniques d'appariement de squelettes ont été proposées dans le contexte de la reconnaissance de formes. Sundar *et al.* [SSGD03] proposent une méthode permettant d'associer une signature à un squelette, pour rechercher les objets les plus proches, dans une base de données. Cette méthode ne s'applique pas à notre cas, car aucune notion d'appariement des branches des squelettes n'y est présente. Bai *et al.* [BL08] suggèrent une approche basée sur l'appariement des extrémités de deux squelettes : l'idée est de comparer tous les chemins allant d'une extrémité à une autre dans chacun des squelettes, pour les mettre en correspondance. Xu *et al.* [XWLB] étendent cette méthode en

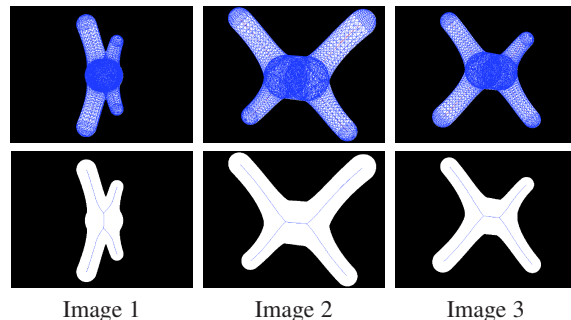


Figure 3: Trois prises de vue d'un même objet 3D (première ligne), et leur squelette perspectif (seconde ligne). On voit que la topologie du premier squelette perspectif est différent des topologies des deuxième et troisième squelettes.

appariant aussi les noeuds internes aux squelettes. Ces méthodes se rapprochent plus de ce que nous voulons ici : cependant, nous ne pouvons les utiliser, car les comparaisons des chemins ne sont possibles que si les squelettes appariés n'ont pas des proportions trop différentes. Or, dans notre cas, la projection perspective peut mener à des différences de proportion notables.

Une autre classe de méthodes d'appariement existe, basée sur représentation particulière du squelette, appelée *shock-graph*. Cette méthode d'appariement a été introduite par Siddiqi *et al.* [SSDZ99]. L'idée est d'appliquer une succession de transformations pour aller d'un squelette orienté à un autre : contracter ou insérer une arête, supprimer ou ajouter une extrémité. Le nombre d'édits est appelée la distance d'édition, et l'estimer est NP-complet [ZSS92] : pour cela, on utilise un algorithme récursif, testant toutes les transformations possibles pour passer d'un squelette à l'autre. Par la suite, Klein *et al.* [KTSK00] ont proposé une représentation des *shock-graph* sous la forme de chaînes d'Euler, pour faciliter leur appariement. Enfin, Sebastian *et al.* [SKK04] étoffent la méthode afin d'apparier les branches de façon plus précise, et réduisent la complexité de l'algorithme d'appariement en utilisant des hypothèses supplémentaires pour éviter d'évaluer une partie des choix. Mis à part la complexité de l'algorithme, cette méthode d'appariement possède la contrainte de ne pouvoir apparier que deux squelettes, et ne permet donc pas de proposer une topologie proche d'un ensemble de squelettes.

Dans la suite, nous utilisons la distance d'édition pour estimer le squelette le plus proche d'un ensemble de squelettes. Contrairement aux méthodes présentées précédemment, nous ne cherchons pas à rechercher un squelette dans une base de données, mais à estimer une topologie cohérente avec un ensemble de squelettes. Pour ce faire, nous utilisons une information supplémentaire, qui est l'appariement des extrémités. Ceci nous permet de plus de diminuer la complexité du problème.

3.2. Positionnement du problème et hypothèses, sur un exemple

Ainsi, nous possédons en entrée un ensemble de n images calibrées d'un même objet, et les squelettes perspectifs correspondants. Nous supposons de plus que l'objet est connexe et acyclique, et qu'il est représentable par un squelette curviligne : il est ainsi l'union d'un ensemble de surfaces canal. De plus, pour éviter une complexité trop élevée, les appariements des extrémités des squelettes sont connus (l'utilisateur les aura associés en les cliquant).

En se basant sur la distance d'édition entre deux squelettes, nous proposons une méthode pour estimer le squelette ayant la topologie la plus proche de l'ensemble des squelettes perspectifs. L'idée est donc de trouver la topologie du squelette, telle que le nombre de transformations pour aller jusqu'à n'importe quel squelette perspectif soit minimal. Pour cela, nous décrivons uniquement les arêtes de chaque squelette, à partir de l'information des extrémités. Ceci nous permet de donner une signature des arêtes, reconnaissable sur chaque squelette, et va nous permettre de choisir les arêtes conservées pour le squelette final : pour minimiser le nombre total de transformations depuis le squelette final jusqu'à chacun des squelettes, nous conservons uniquement les arêtes présentes dans la moitié plus un des squelettes.

De plus, nous représentons les squelettes par des graphes tels que les sommets de chaque graphe correspondent aux jonctions ou aux extrémités de chaque squelette, et tels que les arêtes correspondent aux branches des squelettes. De fait, ces graphes ne comportent pas de sommets de degré 2. Formellement, le graphe $G_i = (V_i, E_i)$ obtenu correspond à la topologie du squelette sur l'image i , qui est connexe et acyclique par hypothèse.

3.3. Illustration de l'appariement

Supposons que nous connaissons les associations des $m = 4$ extrémités (Fig. 3), et que nous avons $n = 3$ graphes (Fig. 3, représentés dans la Fig. 4). Nous allons tout d'abord donner une intuition de notre méthode pour trouver le graphe le plus proche des graphes G_i de la Figure 4 puis la formaliser.

Le principe utilisé se décline en deux étapes : tout d'abord, on construit un graphe commun G_c contenant le même nombre d'extrémités et un unique sommet central, ensuite on ajoute à ce graphe les arêtes nécessaires pour minimiser le nombre total de transformations nécessaires pour obtenir les autres graphes. Ainsi, construisons G_c , un graphe avec m extrémités et un sommet central, relié à toutes les extrémités (cf. Fig. 4). On peut observer que l'insertion de l'arête e_{13} au graphe G_c donne un graphe isomorphe au graphe G_1 . L'insertion de l'arête e_{23} (resp. e_{33}) donne un graphe isomorphe au graphe G_2 (resp. G_3). Puisqu'il y a au total 3 insertions d'arêtes (une pour chaque graphe), le coût total d'édition est de 3 pour G_c .

Il est possible de simplifier les deux insertions similaires

dans les deuxième et troisième graphes. Pour ce faire, nous modifions le graphe G_c , en insérant l'arête e_{23} . G_c est à présent isomorphe aux graphes G_2 et G_3 , et aucune édition n'est nécessaire pour obtenir ces graphes. Pour obtenir le graphe G_1 à partir du graphe G_c , deux opérations sont nécessaires : d'abord, retirer l'arête e_{23} , puis insérer l'arête e_{13} . Nous avons donc à présent un coût d'édition de 2, ce qui est le minimum, puisqu'il n'y a plus de simplifications possibles.

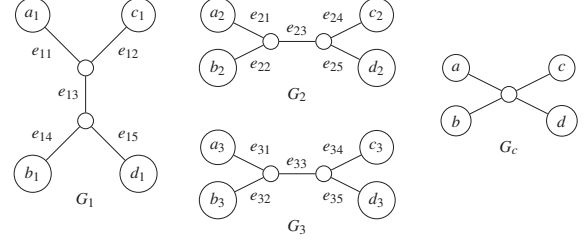


Figure 4: G_1 , G_2 , G_3 : Topologie des trois squelettes perspectifs de la Figure 3. Chaque $(a_i)_{i=1,2,3}$, $(b_i)_{i=1,2,3}$, $(c_i)_{i=1,2,3}$ et $(d_i)_{i=1,2,3}$ sont appariés. G_c : exemple d'un graphe simple initial, correspondant aux G_i .

3.4. Formalisation de l'approche par graphe

Nous introduisons ici une notation pour caractériser les arêtes, permettant de les reconnaître sur les différents squelettes perspectifs.

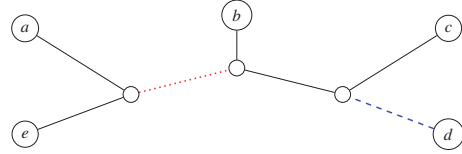


Figure 5: Visualisation de la notation introduite. L'arête rouge en pointillés sépare le graphe en deux parties : la première contient a et e , et la seconde contient b , c et d . On la nomme donc $\{\{a, e\}, \{b, c, d\}\}$. De façon similaire, l'arête bleue en tirets est nommée $\{\{d\}, \{a, b, c, e\}\}$. Notons que $\{a, e\} \subset \{a, b, c, e\}$ et que $\{d\} \subset \{b, c, d\}$.

Notation des arêtes : Comme chaque graphe est acyclique et connexe, chaque arête sépare le graphe en deux parties, la première, $G(A_1)$, contenant l'ensemble des extrémités A_1 , et la seconde, $G(A_2)$, contenant l'ensemble complémentaire A_2 . Nous caractérisons donc chaque arête par la paire $\{A_1, A_2\}$ (cf. Fig. 5), avec :

$$A_1 \cup A_2 = V_e \text{ et } A_1 \cap A_2 = \emptyset$$

où V_e est l'ensemble des extrémités du graphe G_c (qui sont les sommets de degré 1). Cette notation est unique puisqu'il n'y a pas de sommets de degré 2. Nous notons S_{e_i} l'ensemble des paires $\{A_1, A_2\}$ qui caractérisent les arêtes du graphe G_i .

Compatibilité des arêtes : Considérons deux arêtes arbitraires de S_{e_i} , $\{A_1, A_2\}$ et $\{B_1, B_2\}$. Nous avons la propriété suivante (cf. Fig. 5) :

$$A_1 \subsetneq B_1 \text{ et } B_2 \subsetneq A_2 \quad (1)$$

quitte à ré-indicer les ensembles. (Preuve en Annexe)

Adjacence des arêtes : Les deux arêtes partagent un sommet si et seulement si :

$$\forall \{C_1, C_2\} \in S_{e_i}, A_1 \subset C_1 \subset B_1 \Rightarrow C_1 \in \{A_1, B_1\}$$

puisque $C_1 \notin \{A_1, B_1\}$ signifierait que le sous-graphe contenant A_1 est strictement contenu dans le sous-graphe contenant C_1 , ce qui mènerait à une non adjacence des arêtes $\{A_1, A_2\}$ et $\{B_1, B_2\}$.

Construction du graphe : Nous montrons à présent comment reconstruire un graphe isomorphe à G_i à partir de l'ensemble S_{e_i} . Pour ce faire, il faut construire l'ensemble des sommets associés aux arêtes de S_{e_i} . Ces sommets se divisent en deux catégories : les sommets extrémités, et les sommets internes, dont nous montrons la construction dans un premier temps. Ensuite, nous devons relier chacun des sommets à partir des données des arêtes contenues dans S_{e_i} .

Pour déterminer un sommet interne du graphe, nous le décrivons par les différents ensembles d'extrémités auxquelles il est relié. Par exemple, dans le graphe G_1 , le sommet à la jonction de e_{11} , e_{12} et e_{13} est décrit par l'ensemble $\{\{a_1\}, \{c_1\}, \{b_1, d_1\}\}$. Soit $e = \{A_1, A_2\}$ une arête de l'ensemble S_{e_i} . Deux sommets peuvent être construits à partir de cette arête. Nous notons $(e_k)_{k=1, \dots, p}$ les arêtes adjacentes à e , avec $e_k = \{B_{k1}, B_{k2}\}$, telles que $B_{k2} \subset A_2$ (Équation (1)). Toutes les arêtes e_k partagent un même sommet v , qui est l'extrémité de e du côté de A_2 . Une partition $(B_{k2})_{k=1, \dots, p}$ de A_2 existe, où $p+1$ est le degré de v . Nous définissons l'ensemble $v_1 = \{A_1, (B_{k2})_{k=1, \dots, p}\}$ comme le sommet commun à toutes les arêtes, qui constitue une partition de V_e . Par conséquent, pour tout k , B_{k1} est l'union de tous les ensembles de v_1 excepté B_{k2} , ce qui signifie que v_1 peut être défini à partir de n'importe laquelle des arêtes de son voisinage.

Finalement, nous pouvons construire le graphe H_i , contenant tous les sommets construits de cette façon, en reliant les sommets entre eux. Deux sommets v_1 et v_2 sont voisins si :

$$\exists (A, B) \in v_1 \times v_2, \{A, B\} \in S_{e_i}.$$

Construisons à présent un isomorphisme f entre G_i et H_i , c'est-à-dire une fonction bijective conservant les relations d'adjacence des arêtes. Tout d'abord, pour chaque extrémité $a \in V_e$, nous posons $f(a) = \{\{a\}, V_e \setminus \{a\}\}$. Ensuite, pour chaque sommet interne $v \in V \setminus V_e$, nous construisons u , la partition de V_e basée sur l'ensemble des arêtes de S_{e_i} contenant v . Par construction, u est dans le graphe H_i , et nous posons $f(v) = u$. Puisque u existe et est unique, f est bijective, et par construction, c'est un isomorphisme.

Trouver la topologie la plus proche : Puisque nous venons de démontrer que l'ensemble S_{e_i} représente le graphe G_i , on peut insérer ou contracter des arêtes en utilisant l'ensemble S_{e_i} . On note $i(e)$ l'insertion de l'arête e , et $c(e)$ la contraction de l'arête e , où e est une partition de V_e , avec $\#e = 2$. Puisque les ensembles ne sont pas ordonnés, les insertions peuvent être faites dans n'importe quel ordre. En se basant sur ceci, nous pouvons insérer dans un ensemble S_e toutes les arêtes qui sont contenues dans au moins $n/2 + 1$ ensembles S_{e_i} .

Nous avons deux propriétés sur S_e , montrées en annexe : premièrement, l'ensemble S_e représente un graphe, et deuxièmement, le graphe G associé à S_e est le graphe le plus proche des G_i .

Pour finir, nous contractons les arêtes de chacun des squelettes qui ne sont pas présentes dans le graphe commun. Ceci permettra d'associer les branches des squelettes perspectifs sur chacune des images, quitte à en représenter certaines par un sommet unique.

4. Utilisation d'un lagrangien pour appairer les branches

À ce stade de la reconstruction nous avons estimé une topologie pour le squelette 3D, et nous avons associé les branches des squelettes perspectifs sur chaque image. Ainsi, pour chaque branche du squelette 3D, on a un ensemble de n branches projectives. Chaque branche étant représentée par une fonction à un paramètre, nous devons mettre en correspondance les paramétrisations des branches projectives pour pouvoir trianguler les branches du squelette 3D. Pour ce faire, nous définissons dans un premier temps une fonctionnelle à minimiser, sous la forme d'un lagrangien, puis la déterminons dans un second temps par intégration numérique.

4.1. Modélisation de l'appariement par un lagrangien

Supposons que nous ayons un ensemble de n branches perspectives curvilignes associées, une pour chaque image. L'ensemble de ces branches va nous permettre d'estimer une branche du squelette 3D, par triangulation. En effet, considérons un point d'une branche d'un squelette perspectif : celui-ci correspond à un cône, représentant l'ensemble des sphères qu'il est possible de projeter en ce point. Ce cône peut être représenté par une droite de \mathbb{R}^4 . En prenant un point sur chaque branche perspective, nous pouvons obtenir un ensemble de droites de \mathbb{R}^4 , dont l'intersection (ou plutôt, le point le plus proche) correspond à une sphère du squelette 3D. La problématique que nous allons aborder ici sera donc de déterminer l'ensemble des appariements des points des branches perspectives permettant de trianguler les sphères du squelette 3D.

Pour ce faire, nous supposons que :

1. Les droites de \mathbb{R}^4 correspondant à un ensemble de

points des branches perspectives qui ne s'intersectent pas (ou plutôt, sont trop éloignées les unes des autres) ne peuvent pas être appariées.

2. L'appariement se fait de façon continue le long de chaque branche perspective : ceci correspond au cas où aucune auto-occlusion locale n'est présente.
3. Nous savons appairer les extrémités de l'ensemble de branches.

Notre problème est alors modélisé par l'estimation d'une géodésique de l'espace des paramétrisations allant d'une extrémité à l'autre.

Dans un premier temps, traduisons les données du problème ainsi que les hypothèses énoncées ci-dessus. Nous modélisons la branche perspective de l'image i par une fonction L_i , qui à chaque $\tau_i \in [0; 1]$ associe une droite de \mathbb{R}^4 . Il s'agira donc de chercher les (τ_1, \dots, τ_n) qui correspondent à un point (x, y, z, r) du squelette 3D. On va donc construire une fonction P qui associe à chaque (τ_1, \dots, τ_n) le point le plus proche de l'ensemble de droites $(L_i(\tau_i))_{i=1, \dots, n}$ aux moindres carrés.

Pour estimer la distance entre les droites perspectives, nous construisons une fonction $D(\tau_1, \dots, \tau_n)$ qui associe la somme des distances carrées de $P(\tau_1, \dots, \tau_n)$ aux droites $(L_i(\tau_i))_{i=1, \dots, n}$, que nous devons donc minimiser. Ensuite, si nous supposons que l'appariement se fait de façon continue le long de chaque branche, nous devons estimer une fonction $\tau : [t_0; t_1] \rightarrow [0; 1]^n$ continue, telle que $P(\tau(t))$ corresponde à la branche résultante du squelette 3D. Enfin, si nous savons appairer les extrémités, cela signifie que nous pouvons poser $P(0, \dots, 0)$ et $P(1, \dots, 1)$ comme étant des points de la courbe (quitte à re-paramétriser les L_i).

Dans un cas idéal, l'appariement τ optimal vérifierait :

$$\int_{t_0}^{t_1} D(\tau(t)) dt = 0 \quad (2)$$

Cependant, cette équation n'est jamais vérifiée, notamment à cause du bruit. Nous pourrions alors simplement estimer τ de façon à minimiser l'intégrale. Cette solution ne convient pas non plus, car la fonction τ trouvée est très bruitée. Nous posons donc une condition supplémentaire, consistant à minimiser la longueur de τ , en fonction d'un paramètre λ . Ceci consiste donc à minimiser la fonctionnelle suivante :

$$\int_{t_0}^{t_1} \lambda \|\tau'(t)\|^2 + D(\tau(t)) dt \quad (3)$$

Nous avons donc le lagrangien suivant :

$$L(t, \tau, \tau') = \lambda \|\tau'(t)\|^2 + D(\tau(t)) \quad (4)$$

4.2. Résolution par intégration numérique

La fonctionnelle (3) est déterminée pour deux images dans [DMC*15b, DMC*15a] par programmation dynamique, *i.e.* par l'algorithme de Dijkstra [Dij71], en discrétisant le domaine. Pour ce faire, chaque branche est discrétisée

en $p = 100$ points, ce qui nous permet de calculer une carte de distances (*cf.* Fig. 7) entre les points des squelettes de taille p^n sommets. Le principe de l'algorithme est de trouver le chemin reliant deux coins opposés de la carte, en minimisant la somme des valeurs parcourues. La complexité de l'algorithme de Dijkstra étant de l'ordre de $(S + A) \times \ln(S)$, avec S le nombre de sommets et A le nombre d'arêtes, nous avons ici une complexité de $(p^n + p(2^n - 1)) \times \ln(p^n)$ (en effet, pour chaque sommet du graphe, celui-ci possède $2^n - 1$ voisins). Pour $n = 2$ images, cette complexité est de l'ordre de 95.10^3 , pour $n = 3$ images, de l'ordre de 14.10^6 , et pour $n = 4$ images, celle-ci est de l'ordre de 2.10^9 . Un facteur multiplicatif de l'ordre de 140 lie les différentes complexités, ce qui est corroboré par les temps de calculs obtenus : de l'ordre de la seconde pour $n = 2$, de quelques minutes pour $n = 3$, et ne se terminant pas pour $n = 4$. Cette méthode n'est donc pas adaptée pour un nombre d'images trop important. De plus, cette méthode ne permet pas de tenir compte de la dérivabilité de la fonction τ .

Nous allons donc ici résoudre la fonctionnelle (3) en utilisant la méthode du lagrangien discret, qui considère τ comme une trajectoire géodésique, allant de $(0, \dots, 0)$ à $(1, \dots, 1)$, et respectant le lagrangien (4). Pour tout instant t , on estime $\tau''(t)$, $\tau'(t + \Delta t)$ puis $\tau(t + \Delta t)$, tels que :

$$\begin{aligned} 2\lambda \tau''(t) &= \nabla D(\tau(t)) \\ \tau'(t + \Delta t) &= \tau'(t) + \tau''(t) \Delta t \\ \tau(t + \Delta t) &= \tau(t) + \tau'(t) \Delta t \end{aligned} \quad (5)$$

avec Δt un pas de temps faible. La trajectoire est entièrement décidée à partir d'une vitesse et d'une position de départ. Les coordonnées de la position de départ étant connues, ici $(0, \dots, 0)$, il suffit donc de déterminer une vitesse de départ pour que la trajectoire τ passe par le point de coordonnées $(1, \dots, 1)$. Pour ce faire, on arrête l'estimation de la trajectoire dès que $\|\tau(0)\tau(t)\|_\infty \geq 1$, et on estime la distance entre $\tau(t)$ et le point de coordonnées $(1, \dots, 1)$. Cette distance sera minimisée, pour déterminer la vitesse initiale optimale.

4.3. Estimation du paramètre λ

Le dernier point à aborder concernant l'estimation de l'appariement des paramétrisations des branches est la gestion du paramètre λ . Ce choix se fait en posant un λ faible dans un premier temps, que nous incrémentons jusqu'au moment où la géodésique correspondante passe par le point de coordonnées $(1, \dots, 1)$.

Ce paramètre gère la longueur de la courbe : plus il est élevé, plus la courbe est courte. En effet, à l'infini, on a $\tau(t) = (t, \dots, t)$, car $\tau''(t) = \vec{0}$, d'après l'Équation (5). En posant un λ faible dans un premier temps, nous nous assurons donc d'obtenir une solution proche de la solution de l'Équation (2).

Nous observons de plus que si λ est trop faible, le point de coordonnées $(1, \dots, 1)$ ne se situe pas sur la trajectoire de la géodésique. À l'inverse, si λ tend vers l'infini, la géodésique contiendra nécessairement ce point. En incrémentant au fur et à mesure λ par un pas faible (empiriquement, nous partons de $\lambda_0 = 0.2$, que nous incrémentons par pas de 0.1), nous nous assurons donc de trouver la valeur la plus faible de λ , qui permette de faire passer la géodésique à proximité du point de coordonnées $(1, \dots, 1)$. Nous arrêtons l'algorithme quand la géodésique est suffisamment proche de ce point, ce qui est déterminé par un seuil de distance (fixé empiriquement à 0.01).

5. Conditionnement et direction des points de vue

Dans cette section, nous présentons une méthode permettant d'estimer la qualité des points de vue choisis par l'utilisateur. En effet, il existe des cas où l'appariement des branches est mal posé. Après avoir présenté l'influence du choix des points de vue sur la reconstruction, nous proposons une méthode permettant d'estimer si les vues choisies induisent une ambiguïté sur la reconstruction. Nous proposons de plus un choix possible pour une nouvelle vue à acquérir, dans le cas où le problème serait mal posé.

5.1. Présentation du problème

Afin d'apparier les squelettes perspectifs sur les images, nous mettons en correspondance les paramétrisations des n squelettes étudiés. Cependant, il existe des cas où cet appariement est ambigu. Par exemple, nous voyons dans la Figure 6 que cette reconstruction est déformée. Cependant, après projection de cette solution sur les deux vues étudiées, on peut constater que ce squelette correspond parfaitement aux données initiales. Le problème est donc ici mal posé, ce qui induit cette déformation.

Le fait que ce problème soit mal posé est confirmé par l'étude de la carte de distance associée. Nous constatons sur la première carte de distance de la Figure 7 que pour chaque valeur des paramètres, la distance entre les droites correspondantes est très faible, contrairement à une carte de distance mieux conditionnée. Nous venons de mettre en évidence un problème d'ambiguïté sur un cas simple de squelette. Il faut cependant garder à l'esprit que cette ambiguïté peut apparaître dans des squelettes plus complexes que celui que nous avons présenté. Dans la section suivante, nous caractérisons les cas où cette ambiguïté apparaît.

5.2. Caractérisation de l'ambiguïté

Sur la Figure 6, l'ambiguïté provient du fait que le squelette est contenu dans le plan formé par les deux directions de projection. Ainsi, pour chaque point du squelette de la première image, nous pouvons associer tous les points du squelette de la seconde image, et inversement. Nous généralisons

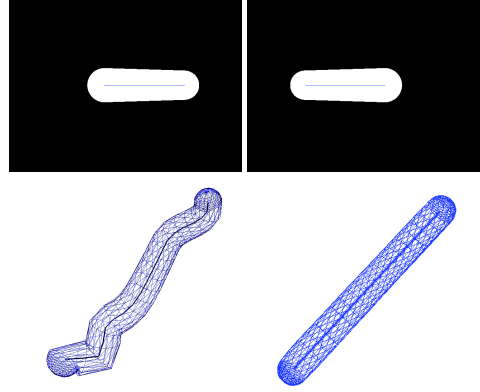


Figure 6: Exemple de l'effet d'un appariement ambigu sur une reconstruction. Les deux images correspondent à une projection orthographique d'un même objet, selon deux plans de projection orthogonaux $((\vec{O}_x, \vec{O}_y)$ pour la première image et (\vec{O}_z, \vec{O}_y) pour la seconde). Deux reconstructions possibles sont présentées, pour une valeur du paramètre λ faible (à gauche), et pour une valeur du paramètre élevée (à droite).

cette observation, pour caractériser mathématiquement cette ambiguïté : l'ambiguïté est présente lorsque l'espace vectoriel formé par les directions de projection et les tangentes aux squelettes projectifs est de dimension 2.

Partons d'un cas simple, en supposant que l'ensemble des centres du squelette 3D est défini par un segment $C(s)$ de \mathbb{R}^3 . Nous supposons que nous avons deux caméras orthographiques, ayant pour direction de projection respectives \vec{d}_1 et \vec{d}_2 . En projetant orthogonalement le squelette selon la direction \vec{d}_i , on obtient le squelette $C_i(s_i)$ tel que :

$$C_i(s_i) = C_{i0} + s_i \vec{T}_i, s_i \in [0, 1]$$

avec C_{i0} la projection de C_0 sur le plan de projection orthogonal à \vec{d}_i , et \vec{T}_i la projection du vecteur \vec{T} sur ce plan. $\vec{C}_{i0}\vec{C}_0$ et \vec{d}_i sont ainsi colinéaires, et \vec{T}_i et \vec{d}_i sont orthogonaux.

Considérons l'opération de rétroprojection du squelette $C_i(s_i)$. L'ensemble des points se projetant sur $C_i(s_i)$ est compris dans l'ensemble :

$$\left\{ Q \in \mathbb{R}^3, \exists (k_i, s_i) \in \mathbb{R}^+ \times [0, 1], Q = C_{i0} + k_i \vec{d}_i + s_i \vec{T}_i \right\}.$$

Ce qui implique que $\vec{C}_{i0}Q$ appartient à l'espace vectoriel $\vec{\mathcal{H}}_i$ engendré par $\left\{ \vec{d}_i, \vec{T}_i \right\}$. Par définition, $\vec{C}_{i0}C(s)$ appartient

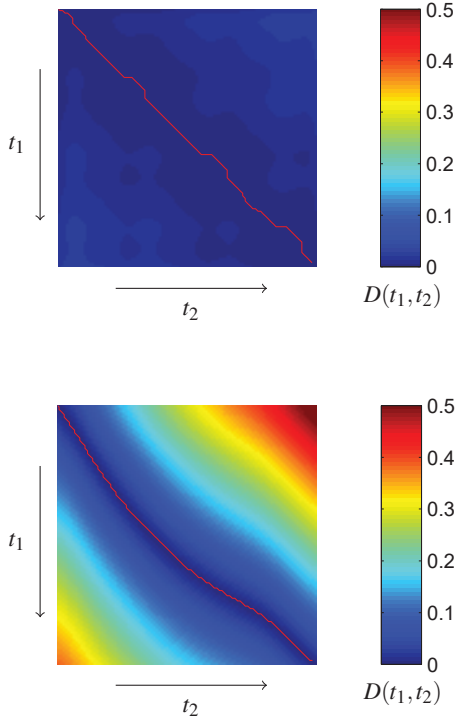


Figure 7: Cartes de distance utilisées pour l'appariement de deux squelettes différents. La première carte de distance correspond aux squelettes perspectifs de la Figure 6, où l'appariement est ambigu. La seconde carte de distance correspond à un cas idéal de reconstruction, où le minimum est atteint dans une vallée.

à cet espace vectoriel. Nous avons de plus :

$$\overrightarrow{C_{i0}C(s + \Delta s)} \simeq \overrightarrow{C_{i0}C(s)} + \Delta s \overrightarrow{C'(s)}$$

d'où $\overrightarrow{C'(s)} \in \overrightarrow{\mathcal{H}_i}$.

Nous allons à présent caractériser l'ambiguïté décrite. Soit un espace vectoriel $\overrightarrow{\mathcal{H}_i}$ de \mathbb{R}^3 associé à chaque image, l'intersection $\overrightarrow{\mathcal{H}_1} \cap \overrightarrow{\mathcal{H}_2}$ de deux espaces vectoriels est de dimension 1 au minimum, car la direction de la tangente au squelette $\overrightarrow{C'(s)}$ appartient aux deux, comme nous venons de le montrer.

Notons aussi que, de par le fait que chaque espace vectoriel est de dimension 2, la dimension maximale de leur intersection est 2. Et c'est justement ici qu'apparaît l'ambiguïté : si l'intersection est de dimension 2, la triangulation ne donne pas une unique solution.

Pour estimer l'ambiguïté d'une solution, il faut donc vérifier que $\overrightarrow{\mathcal{H}_1} \cap \overrightarrow{\mathcal{H}_2}$ n'est pas de dimension 2, donc vérifier que

$\overrightarrow{\mathcal{H}_1} + \overrightarrow{\mathcal{H}_2}$ n'est pas de dimension 2. De fait, si la dimension de $\overrightarrow{\mathcal{H}_1} + \overrightarrow{\mathcal{H}_2}$ est 3, alors la dimension de $\overrightarrow{\mathcal{H}_1} \cap \overrightarrow{\mathcal{H}_2}$ est 1.

Soit la famille $\{\overrightarrow{d_1}, \overrightarrow{T_1}, \overrightarrow{d_2}, \overrightarrow{T_2}\}$, pour évaluer la dimension de l'espace engendré par celle-ci, il suffit d'étudier le rang de la matrice A formée par ces différents vecteurs (normalisés). Or, le rang d'une matrice étant caractérisé par son conditionnement, nous évaluons donc le conditionnement de A , qui correspond au rapport entre la première et la troisième valeur propre de cette matrice $\frac{\lambda_1}{\lambda_3}$. Plus cette valeur est faible, plus le conditionnement est bon, l'idéal étant un conditionnement de 1.

Pour un squelette non linéaire, modélisé par une fonction dérivable C , nous considérons ses projections C_i . Pour chaque valeur du paramètre s_i , nous approximations le squelette au point $C_i(s_i)$ par une droite, telle que :

$$C_i(s_i + \delta) = C_i(s_i) + \delta \overrightarrow{C'_i(s_i)} + o(\delta)$$

avec $o(\delta)$ une fonction vectorielle de norme faible quand $|\delta|$ est faible. Soit $\overrightarrow{d_i}$ la direction de projection (perspective ou orthographique) au point $C_i(s_i)$. Nous devons étudier ici de la même façon que précédemment la dimension de la famille $\{\overrightarrow{d_1}, \overrightarrow{d_2}, \overrightarrow{C'_1(s_1)}, \overrightarrow{C'_2(s_2)}\}$, afin de déterminer si le problème est localement bien conditionné ou non, en observant le rapport des première et troisième valeurs singulières de la matrice A associée.

Remarquons que l'estimation des valeurs singulières de la matrice A nous fournit une base orthonormée de \mathbb{R}^3 , dont chacun des vecteurs \vec{v}_1 , \vec{v}_2 et \vec{v}_3 correspond aux valeurs singulières respectives λ_1 , λ_2 et λ_3 . Si le rapport entre λ_1 et λ_3 est trop élevé, nous pouvons déduire la direction d'une nouvelle vue : il suffit de prendre le vecteur \vec{v}_3 .

Enfin, pour un nombre d'images n plus grand, la construction de la matrice A de conditionnement est exactement la même, et elle possédera $2n$ colonnes. De la même façon, le conditionnement de la reconstruction de la branche est donné par $\frac{\lambda_1}{\lambda_3}$. Nous pouvons également estimer la direction de projection privilégiée pour minimiser le conditionnement par \vec{v}_3 .

5.3. Évaluation du conditionnement sur images de synthèse

Nous montrons ici un ensemble d'exemples sur des squelettes curvilignes à une seule branche. Les deux caméras sont perspectives, et placées telles que leurs plans de projection soient orthogonaux. Pour chaque exemple, nous évaluons la carte de distance associée aux squelettes projectifs. L'appariement des paramètres est ensuite effectué, et montré sur la carte de distance. Enfin, nous estimons le conditionnement de la solution le long de la courbe trouvée.

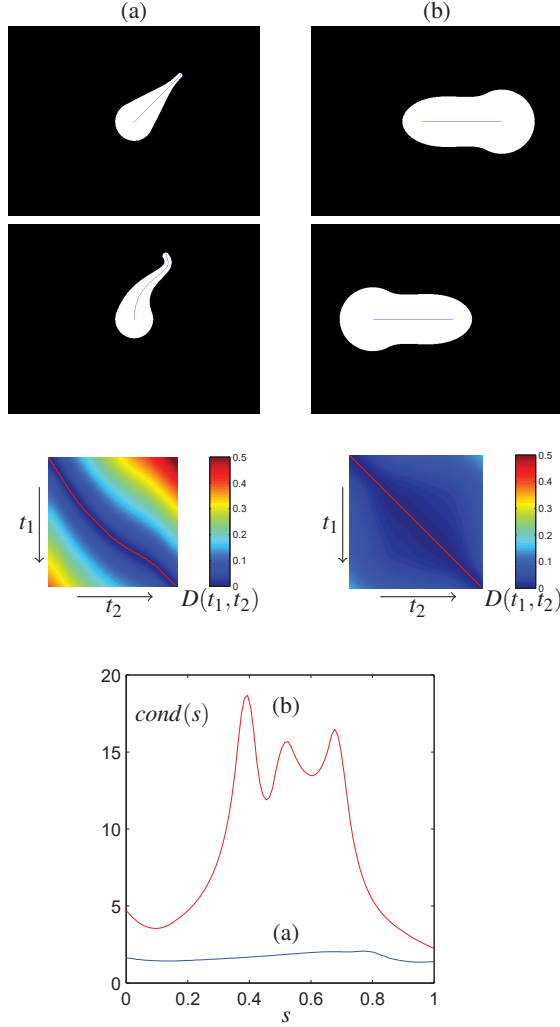


Figure 8: Estimation du conditionnement sur deux squelettes. En haut sont présentées les segmentations des images acquises et ainsi que leurs squelettes estimés en 2D, au milieu, la carte de distance avec l'appariement estimé, et en bas, le conditionnement le long de la mise en correspondance (la courbe bleue représente le conditionnement pour l'exemple (a), et la courbe rouge représente le conditionnement pour l'exemple (b)). L'exemple (a) présente un cas idéal, sans ambiguïté. L'exemple (b), quant à lui, présente une forte ambiguïté sur une partie de la courbe. En effet, nous observons au milieu de la courbe de conditionnement que celui-ci est élevé. Ceci correspond à un plateau de la fonction de distance, et indique donc que plusieurs possibilités existent pour appairer les paramétrisations des squelettes

Nous observons sur l'exemple (a) de la Figure 8 que le chemin minimal est visible sur la carte de distance, sous la forme d'une vallée de couleur bleue. Le conditionnement est ici faible, aux alentours de 2 au maximum. En revanche, sur l'exemple (b), la vallée bleue de la carte de distance est plus étendue : il est plus difficile de distinguer le chemin minimal ici. Le conditionnement s'en ressent : il est plus élevé au milieu de la courbe. Notons qu'il est supérieur à 2 tout le long de la courbe. Pour le diminuer, il faudra donc choisir une nouvelle vue, comme indiqué dans la section précédente.

6. Évaluation de la reconstruction sur images de synthèse et réelles

Nous évaluons quantitativement notre algorithme sur des squelettes de synthèse, puis qualitativement sur images réelles. Pour ce faire, nous acquérons un ensemble d'images calibrées de chaque objet, choisies judicieusement. Ensuite, nous estimons le squelette perspectif associé à chaque projection de l'objet 3D (pour les images réelles, la silhouette est préalablement segmentée à partir de l'algorithme Grabcut [RKB04]). L'étape suivante est celle de l'association des branches de squelette, pour estimer la topologie du squelette 3D. Enfin, nous triangulons chacune des branches du squelette 3D, et estimons la surface associée à partir des équations d'une surface canal [PP97]. Comme attendu, nous notons que pour les images de synthèse, l'erreur relative au maillage original diminue quand le nombre d'images augmente (cf. Tab. 1).

Pour évaluer la validité de la topologie estimée pour le squelette reconstruit, nous testons notre algorithme sur un ensemble de 7 objets réels. Deux résultats sont présentés sur la Figure 9. Notons que notre méthode reconstruit un objet complet et fermé à partir des images calibrées et quelques interactions (les appariements des extrémités). La tête et les bras sont bien reconstruits, mais le corps de la peluche bleue n'est pas aussi plat que celui de la vraie peluche. Ceci est dû au fait que notre méthode ne tient pas compte des squelettes surfaciques. De plus, comme la topologie est la plus proche de l'ensemble des squelettes perspectifs, quelques asymétries peuvent apparaître sur la topologie estimée, par exemple pour les bras, qui ne sont pas connectés au même sommet du tronc (cf. Fig. 9).

7. Conclusion

Dans cet article, nous avons présenté un algorithme permettant d'effectuer une reconstruction basée squelette d'un objet, à partir d'un nombre de vues plus grand que 2. Pour ce faire, nous avons tout d'abord introduit une méthode s'appuyant sur une représentation par graphe permettant d'appairer les branches des différents squelettes perspectifs, pour estimer la topologie du squelette reconstruit. Ensuite, nous avons présenté une méthode de mise en correspondance des branches sur les différentes vues, en modélisant le problème

Nombre d'images	2	3	4	5
Squelette 1	2.44% 2.11	2.13% 2.32	1.49% 1.95	1.57% 2.36
Squelette 2	4.87% 2.84	3.34% 2.02	2.46% 1.74	1.90% 1.94
Squelette 3	4.97% 4.78	4.44% 4.10	4.07% 2.72	3.99% 2.77
Squelette 4	4.68% 4.06	3.57% 3.05	3.51% 2.53	3.36% 2.81

Table 1: Reconstruction de 4 squelettes composés, avec respectivement 4, 4, 5 et 6 extrémités. En utilisant 2, 3, 4 et 5 images calibrées pour chaque objet, l'erreur est la distance de Hausdorff entre le maillage originale et sa reconstruction, normalisée par la diagonale de la boîte englobante. En dessous, est donné le conditionnement maximal pour chaque squelette reconstruit.

sous la forme de l'estimation d'une géodésique. Enfin, nous avons proposé une méthode d'évaluation du conditionnement de cette mise en correspondance, permettant aussi d'estimer la prochaine vue à ajouter pour obtenir une reconstruction plus précise.

Plusieurs pistes apparaissent pour étendre ces travaux : tout d'abord, il semble nécessaire d'étendre l'algorithme d'appariement des squelettes perspectifs en prenant en compte des topologies contenant des cycles, et un appariement partiel des extrémités. En effet, pour l'instant notre technique est limitée aux objets ne possédant pas de trous, et dont les extrémités sont visibles sur toutes les images.

Ensuite, la deuxième piste à explorer est la correction de la surface canal triangulée, pour en éviter les auto-intersections. En effet, les phénomènes d'auto-intersection locale de la surface canal sont caractérisables localement, en étudiant les dérivées des fonctions $C(t)$ et $r(t)$, représentant respectivement le centre et le rayon. Par la méthode d'estimation de la trajectoire que nous utilisons, nous avons accès à ces données, ce qui nous donne la possibilité de corriger localement les auto-intersections de la surface canal.

Enfin, la dernière piste à étudier consiste à prendre en compte les squelettes surfaciques. Pour ce faire, la fonction de triangulation doit être modifiée, pour trianguler non pas des sphères, mais des ellipses.

8. Annexe

Compatibilité des arêtes d'un graphe

Pour tout couple d'arêtes $\{A_1, A_2\}$ et $\{B_1, B_2\}$ d'un même graphe représenté par un ensemble S_e , on a la propriété suivante :

$$A_1 \subsetneq B_1 \text{ et } B_2 \subsetneq A_2 \quad (6)$$

quitte à ré-indicer les ensembles.

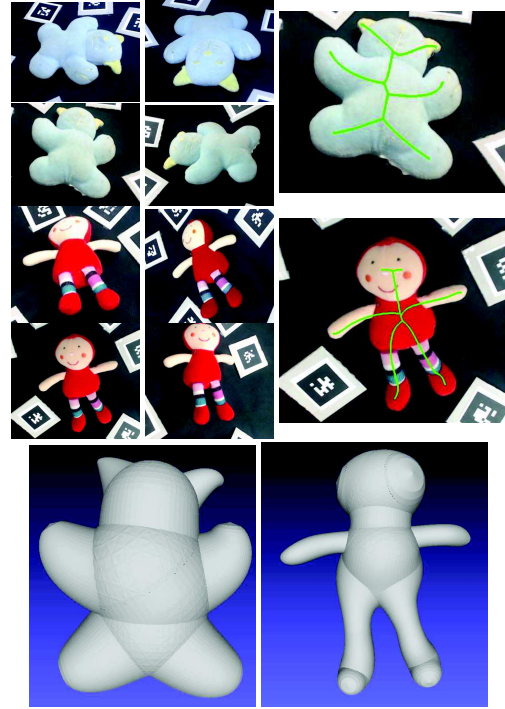


Figure 9: Reconstruction de deux peluches en utilisant notre méthode. 4 images acquises sont présentées pour chaque peluche. À droite, le squelette 3D est projeté sur l'une des images. En bas, les reconstructions sont obtenues pour chaque peluche.

Comme les squelettes sont acycliques et connexes, chaque arête sépare le graphe G en deux sous-graphes. Sans perte de généralité, supposons que l'arête $\{B_1, B_2\}$ est dans $G(A_2)$, et que l'arête $\{A_1, A_2\}$ est dans $G(B_1)$. Supposons aussi que $B_2 \not\subseteq A_2$, donc que $G(B_2)$ n'est pas inclus dans $G(A_2)$. Comme $\{B_1, B_2\}$ est dans $G(A_2)$, il y a un sommet v qui est dans $G(B_2)$ et $G(A_2)$. Par notre supposition, on peut en déduire qu'il y a un sommet u qui est dans $G(B_2)$ mais pas dans $G(A_2)$, donc u est dans $G(A_1)$. $G(B_2)$ est connexe, donc il contient un chemin allant de u à v . Comme G est acyclique, ce chemin est unique, et il contient l'arête $\{A_1, A_2\}$. Donc cette arête est dans $G(B_2)$, ainsi que dans $G(B_1)$. Cependant par construction, il n'y a pas d'intersection entre ces deux graphes, ce qui signifie que l'Équation (6) est vérifiée. \square

Cohérence et proximités aux graphes de l'ensemble S_e des arêtes

Nous devons prouver deux assertions : premièrement, que l'ensemble S_e représente un graphe, et deuxièmement, que le graphe G associé à S_e est le graphe le plus proche des G_i . La deuxième assertion est prouvée par le fait que toutes les insertions sont commutatives. En ce qui concerne la première assertion, supposons que nous ayons deux arêtes in-

compatibles e_1 et e_2 dans S_e , i.e. que l'Équation (1) n'est pas respectée. Ces arêtes proviennent des graphes G_i , et sont contenues dans au moins $n/2 + 1$ ensembles S_{e_i} . Comme il n'y a que n graphes, il existe un graphe G_i qui contient e_1 et e_2 , ce qui signifie que ces arêtes sont compatibles, il y a donc contradiction.

Remerciements : ce travail a été partiellement financé par le CIMI (Centre International de Mathématiques et d'Informatique), programme d'excellence ANR-11-LABX-0040-CIMI du programme ANR-11-IDEX-0002-02.

Références

- [AHL07] AUJAY G., HÉTROUY F., LAZARUS F., DEPRAZ C. : Harmonic skeleton for realistic character animation. In *Symposium on Computer Animation* (2007), pp. 151–160.
- [BL08] BAI X., LATECKI L. J. : Path similarity skeleton graph matching. *IEEE Transactions on PAMI*. Vol. 30, Num. 7 (2008), 1282–1292.
- [Blu67] BLUM H. : A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form* (1967).
- [CBK03] CHEUNG K. M. G., BAKER S., KANADE T. : Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *IEEE CVPR conference* (2003), vol. 1, pp. 77–84.
- [CG06] CAGLIOTI V., GIUSTI A. : Reconstruction of canal surfaces from single images under exact perspective. In *ECCV 2006*. 2006, pp. 289–300.
- [CZS*13] CHEN T., ZHU Z., SHAMIR A., HU S.-M., COHEN-OR D. : 3-sweep : Extracting editable objects from a single photo. *ACM Transactions on Graphics*. Vol. 32, Num. 6 (2013), 195–204.
- [Dij71] DIJKSTRA E. W. : *A short introduction to the art of programming*, vol. 4. Technische Hogeschool Eindhoven Eindhoven, 1971.
- [DMC*15a] DURIX B., MORIN G., CHAMBON S., ROUDET C., GARNIER L. : Caractérisation de la projection du squelette d'une surface canal 3D : Application à la reconstruction 3d à partir de deux images, Avril 2015.
- [DMC*15b] DURIX B., MORIN G., CHAMBON S., ROUDET C., GARNIER L. : Towards skeleton based reconstruction : From projective skeletonization to canal surface estimation. In *International Conference on 3D Vision* (2015), pp. 545–553.
- [FP10] FURUKAWA Y., PONCE J. : Accurate, dense, and robust multi-view stereopsis. *IEEE Transactions on PAMI*. Vol. 32, Num. 8 (2010), 1362–1376.
- [KB99] KATO H., BILLINGHURST M. : Marker tracking and HMD calibration for a video-based augmented reality conferencing system. *International Workshop on Augmented Reality* (1999), 85–94.
- [KCZO07] KAVAN L., COLLINS S., ZARA J., O'SULLIVAN C. : Skinning with dual quaternions. In *Symposium on Interactive 3D Graphics and Games* (2007), pp. 39–46.
- [KTSK00] KLEIN P., TIRTHAPURA S., SHARVIT D., KIMIA B. : A tree-edit-distance algorithm for comparing simple, closed shapes. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms* (2000).
- [LGS12] LIVESU M., GUGGERI F., SCATENI R. : Reconstructing the curve-skeletons of 3d shapes using the visual hull. *IEEE Transactions on Visualization and Computer Graphics*. Vol. 18, Num. 11 (2012), 1891–1901.
- [LS13] LIVESU M., SCATENI R. : Extracting curve-skeletons from digital shapes using occluding contours. *The Visual Computer*. Vol. 29, Num. 9 (2013), 907–916.
- [LWFCML03] LIU P.-C., WU F.-C. AND MA W.-C., LIANG R.-H., OUHYOUNG M. : Automatic animation skeleton using repulsive force field. In *Pacific Graphics* (2003), pp. 409–413.
- [PP97] PETERNELL M., POTTSMANN H. : Computing rational parametrizations of canal surfaces. *Journal of Symbolic Computation*. Vol. 23, Num. 2-3 (1997), 255–266.
- [RKB04] ROTHER C., KOLMOGOROV V., BLAKE A. : Grabcut : Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*. Vol. 23, Num. 3 (2004), 309–314.
- [SKK04] SEBASTIAN T. B., KLEIN P. N., KIMIA B. B. : Recognition of shapes by editing their shock graphs. *IEEE Transactions on PAMI*. Vol. 26, Num. 5 (2004), 550–571.
- [SSdBA11] SERINO L., SANNITI DI BAJA G., ARCELLI C. : Using the skeleton for 3d object decomposition. In *Image Analysis*, Heyden A., Kahl F., (Eds.), vol. 6688 de *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011, pp. 447–456.
- [SSDZ99] SIDDIQI K., SHOKOUFANDEH A., DICKINSON S. J., ZUCKER S. W. : Shock graphs and shape matching. *IJCV*. Vol. 35, Num. 1 (1999), 13–32.
- [SSGD03] SUNDAR H., SILVER D., GAGVANI N., DICKINSON S. : Skeleton based shape matching and retrieval. In *Shape Modeling International* (2003), pp. 130–139.
- [XWLB] XU Y., WANG B., LIU W., BAI X. : Skeleton graph matching based on critical points using path similarity. In *ACCV 2009*. pp. 456–465.
- [ZSS92] ZHANG K., STATMAN R., SHASHA D. : On the editing distance between unordered labeled trees. *Information Processing Letters*. Vol. 42, Num. 3 (1992), 133–139.